Topic Models Regularization and Initialization for Regression Problems

Evgeny Sokolov
Yandex Data Factory
Leo Tolstoy st., 16
Moscow, Russia
esky@yandex-team.ru

Lev Bogolubsky
Yandex
Leo Tolstoy st., 16
Moscow, Russia
bogolubsky@yandex-team.ru

ABSTRACT

We propose a method for regression over texts that transforms sparse texts to dense features using regularized topic models. We also discuss the problem of topic model initialization, propose a Naive Bayes-based approach and compare it to many other approaches. Our topic model achieves quality comparable to vector space models using only 10 topics. It also outperforms other topic model-based feature generation methods such as PLSA and Supervised LDA.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications; H.3.3 [**Information Storage and Retrieval**]: Information Filtering

Keywords

Topic Modeling, Regularization, Initialization, EM-Algorithm, Naive Bayes

1. INTRODUCTION

Topic modeling is one of the actively developing branches of statistical text analysis. A probabilistic topic model detects the subject-matter of a collection of text documents by describing each topic by a discrete distribution over the set of terms, and each document — by a discrete distribution over the topic set. Topic modeling has become widely used in a series of applied problems, including newsfeed analysis [2], scientific publication analysis [3], blog analysis [4].

Despite the great number of papers on methods for topic model inference [5, 6, 7] and on the applications of topic models to the classification, insufficient attention was paid to the applications of topic models to the regression problems. In most papers devoted to those subjects [8, 9, 10, 11], various modifications of the Latent Dirichlet Analysis (LDA) model are studied. All these models are proposed to be learned by means of Bayesian inference, and as a model becomes more complicated, both the inference and the formulas complexity grow. In addition, the complication of the model does not necessarily lead to the significant improvement [9].

In this work, another approach to solving of the regression problems with topic models is suggested — namely, the usage of the target vector in the context of the Additve Regularization of Topic Models (ARTM) methodology. Specifically, we introduce a regularizer which demands that the target variable is well predicted with a linear formula over the probabilities of document belonging to the topics. The model suggested can be inferred iteration by iteration with the EM-algorithm, which can be easily scaled [14].

The criterion being optimized during the inference of any topic model is a complex and non-convex function which has many local extrema. Therefore the result of an optimization is strongly dependent on the initialization method. In this paper we introduce and compare various ways of the topic model initialization. We show that the proper choice of the initialization method can lead to a quality gain close to double. Moreover, the simultaneous usage of a proper initialization method and a proper regularization method leads to the significant reduction of the number of iterations needed for the EM-algorithm convergence. One can say that the regression topic model adjustment consists of two stages — at first, we take appropriate features, which are suitable for the problem solving, as an initialization for the training set. Then, by means of a topic model we construct a regular way of extraction of those features from the text data.

2. RELATED WORK

Most papers devoted to solving regression problems by the means of topic modeling propose models based on LDA. They allow to use the real-valued target vector while constructing the model. For instance, there is a basic method for solving the regression problems with topic models, a modification of the LDA model called Supervised LDA [8]. This method modifies the generative model for texts with an assumption that the target variable is generated as a linear combination of document topic probabilities with addition of some gaussian noise. This assumption is equivalent to the addition of the MSE regularizer to a loss function. In our method, we incorporate such regularizer to a simpler PLSA model [15] and show that it achieves similar results, but can be optimized by a simpler and faster procedure.

In the paper [9], an improvement of Supervised LDA is proposed — the MedLDA method, introducing a margin maximization requirement. In the paper [11], a method allowing usage of some additional real-valued information while constructing the topic model is proposed. However, its

TM'15, October 19, 2015, Melbourne, Australia.

ACM 978-1-4503-3784-7/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2809936.2809940.

applications for the regression problems are not discussed. Finally, the Inverse Regression Topic Model [10] was introduced recently. In this model, the distribution of words in a topic depends on the document and changes proportionally to the value of the target variable.

A number of initialization methods for EM-algorithm were proposed [12, 13]. However, they are all designed for the unsupervised case and are usually compared based on unsupervised metrics and clustering quality. In this paper we propose initializations that directly use supervised information and compare them based on resulting regression quality.

3. METHOD

Let D denote a collection of texts (documents) and W denote a set of all terms from these texts. Each text $d \in D$ is represented as a sequence of n_d words (w_1, w_{n_d}), where each word belongs to the vocabulary W. We denote the number of occurrences of the word w in the document d by n_{dw} and the total number of words in the document d by n_d.

Assume that each word occurrence in each document corresponds to some latent topic t from a finite set of topics T. We can consider a text collection as a sample of triples (w_i, d_i, t_i), $i = 1, \dots, n$, where documents d_i and words w_i are observed variables and topics t_i are hidden variables. It is assumed in topic modeling that each topic generates terms regardless of the document: $p(w \mid t) = p(w \mid d, t)$. We can then express probability of word occurrence in the document as

$$p(w \mid d) = \sum_{t \in T} p(t \mid d) p(w \mid t).$$

To learn a topic model means to find probabilities $\phi_{wt} = p(w \mid t)$ and $\theta_{td} = p(t \mid d)$ given a collection D. This problem is equivalent to a problem of finding an approximate factorization of sparse counter matrix $F = (n_{dw}/n_d)_{W \times D}$ into dense matrices $\Phi = (\phi_{wt})_{W \times T}$ and $\Theta = (\theta_{td})_{T \times D}$:

$$L(F, \Phi, \Theta) \to \min_{\Phi, \Theta},$$

where L is a log-likelihood function:

$$L(F, \Phi, \Theta) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \phi_{wt} \theta_{td}.$$

Suppose that we have r additional objectives $R_i(F, \Phi, \Theta)$, $i = 1, \dots, r$ called *regularizers*, that have to be optimized together with the likelihood. To solve this multi-objective optimizations problem we can maximize a linear combination of the objectives with nonnegative regularization coefficients τ_i:

$$L(F, \Phi, \Theta) + \sum_{i=1}^{r} \tau_i R_i(F, \Phi, \Theta) \to \min_{\Phi, \Theta}.$$

This objective can be optimized iteratively by the EM-algorithm, see [14] for details. The described framework is called Additive Regularization of Topic Models (ARTM).

We consider the following regularized log-likelihood functional of a topic model:

$$Q(\Phi, \Theta) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \phi_{wt} \theta_{td} - \quad (1)$$
$$- \frac{\lambda}{2} \sum_{d \in D} \left(y_d - \sum_{t \in T} v_t \theta_{td} \right)^2 .$$

We can see that the regularizer here encourages models in which the target variable y_d of each document is well predicted by a linear function of the topic probabilities θ_{td}. Note that the weight vector $v = (v_1, \dots, v_d)$ is also to be learned. We obtain the following steps of the EM-algorithm for optimizing the functional above:

- E-step:
$$H_{dwt} = \frac{\phi_{wt} \theta_{td}}{\sum_{s \in T} \phi_{ws} \theta_{sd}};$$
- M-step:
$$\phi_{wt} \propto \sum_{d \in D} n_{dw} H_{dwt};$$
$$\theta_{td} \propto \sum_{w \in d} n_{dw} H_{dwt} + \lambda v_t \theta_{td} \left(y_t - \sum_{t \in T} v_t \theta_{td} \right);$$
$$v = (\Theta \Theta^T)^{-1} \Theta y.$$

It is well-known that the initial approximation plays a key role in the topic model adjustment. In this paper, we suggest to use an initial approximation such that the usage of the topic probabilities in documents θ_{td} from this approximation as features alone gives us a possibility to obtain a decent quality for the regression. Let us consider several ways of constructing such approximations.

Topic models are usually intitalized in quite simple ways. One of the options [5] is to generate T groups of documents and initialize each topic according to the distribution of words in the corresponding group. By denoting the indices of documents in the i-th group as D_i, we obtain the formulas for initialization:

$$\phi_{wt} \propto \sum_{d \in D_t} \sum_{w \in d} n_{dw};$$
$$\theta_{td} \propto [d \in D_t].$$

There are lots of ways to construct such groups. They may be disjoint or not, initialized randomly or by clustering methods. The group size can be also adjusted.

Another way is to strictly associate each word with a certain topic. Let $h(w)$ be a function mapping a word w to a topic t. Then the topic model is initialized as follows:

$$\phi_{wt} \propto \sum_{d \in D} [w \in d][h(w) = t] n_{dw};$$
$$\theta_{td} \propto \sum_{w \in d} [h(w) = t] n_{dw}.$$

Suppose that $h(w)$ is a hash function with the set of values $T = \{1, \dots, |T|\}$. In this case, such approach is equivalent to hashing kernels [16], which allow us to obtain linear reduction of features space and logarithmic loss of quality. Hashing kernels are widely used in linear methods of machine learning [17]. In this paper, the hash function MurmurHash3 is used.

In the approach described, all occurrences of a certain word in a document are to be associated with the same topic. However, we can change this situation by means of independent generation of a topic for each term in document as a random value. Suppose ξ_{dwi}, $i = 1, \dots, n_{dw}$, are randomly generated topics for all words. In this case, the initialization looks as follows:

$$\phi_{wt} \propto \sum_{d \in D} [w \in d] \sum_{i=1}^{n_{dw}} [\xi_{dwi} = t];$$

$$\theta_{td} \propto \sum_{w \in d} \sum_{i=1}^{n_{dw}} [\xi_{dwi} = t].$$

We propose to use initialization methods that use naive Bayes classifier applied to the sample, combined with various approaches to the target binarization. Consider $|T|$ various thresholds $b_1, \dots, b_{|T|}$ used for the binarization of the target feature — for instance, they can be chosen according to its CDF quantiles. We propose to construct a topic t basing on the threshold b_t. Let us find a fraction of documents from the class $[y_d < b_t]$ among all documents containing the word w:

$$r_{wt} = \frac{\sum_{d \in D} [y_d < b_t][w \in d]}{\sum_{d \in D} [w \in d]}.$$

This value estimates the probability to obtain the class $[y_d < b_t]$ given that the document contains the word w. If it is close to 0.5, the word is weakly correlated with the target vector. If it is close to zero or one, one can assume that there exists some connection between this word and the target vector. Hence, the importance of a word can be measured with the value

$$s^1_{wt} = \max(r_{wt}, 1 - r_{wt})$$

or

$$s^2_{wt} = \max\left(\ln \frac{r_{wt}}{1 - r_{wt}}, \ln \frac{1 - r_{wt}}{r_{wt}}\right).$$

We can also equate the importance to the probability estimate r_{wt}:

$$s^3_{wt} = r_{wt}.$$

We associate the word w with the topic t if its importance exceeds a threshold:

$$\phi_{wt} \propto [s_{wt} > \alpha].$$

We can as well make topic probabilities proportional to the importance estimates:

$$\phi_{wt} \propto s_{wt}.$$

The naive Bayes classifier estimates the probability of the document belonging to the class as a product over all words of probabilities of the given class under condition of the given word. We can initialize the probabilities of topics in documents in the analogous way:

$$\theta_{td} \propto \prod_{w \in d} s_{wt}^{n_{dw}}.$$

It is better to use the logarithmic initialization so that we avoid computational instabilities caused by small values:

$$\theta_{td} \propto \sum_{w \in d} n_{dw} \ln s_{wt}. \tag{2}$$

We can as well estimate θ_{td} in terms of the number of words which have greater importance estimate s_{wt}:

$$\theta_{td} \propto \sum_{w \in d} n_{dw} [s_{wt} > \alpha]. \tag{3}$$

Every way of initialization requires smoothing of the matrices Φ and Θ. We smooth these matrices by adding a positive constant α to each element before normalizing their rows or columns to one.

4. EXPERIMENTS

We have chosen three datasets for the experiments, namely MovieReviews, Salary and Yelp. The sample MovieReview [1] consists of various user reviews of movies, where we attempt to predict the integer-valued rating from 1 to 5 for each of them. The target vector has undergone the transformation $\ln(1 + x)$. The Salary sample is based on the problem from the website `kaggle.com` called “Adzuna Job Salary Prediction”. In this problem, we are to determine a worker's annual salary in Pounds Sterling basing on the text of an advertisement. The Yelp sample is also based on the `kaggle.com` “Yelp Recruiting Competition” dataset. We try to predict the number of stars in the review basing on the text in this problem. We have also applied the $\ln(1 + x)$ transformation to the target values. We have performed words filtering for all of the samples. All words represented in less than 5 documents of the corpus have been deleted, as well as all words represented in more than 25% of documents.

4.1 Initialization methods

Following initialization options have been tested:

1. **random**: the random initialization. Elements of the words distribution in each topic and the topics distribution in each document are generated from the uniform distribution on the segment $[0, 1]$ and normed afterwards.

2. Clustering methods:

   - **hash**: strict association of each word with a topic according to the hash value.
   - **random_topic**: random generation of a topic for each occurrence of a word to a document.
   - **subsample**: topic generation based on average distributions of words in subsamples of documents; the parameter here is the number of documents used used to initialize every topic — the options from the set $\{1, 2, 3, 5\}$ have been considered.
   - **cluster**: similarly to the previous method, but the documents are split into groups by means of the clustering method K-Means with K-Means++ [18] initialization.

3. Bayesian methods. All possible combinations of the following stages have been considered:

   - Words importance computation s_{wt} (**score_type**):
     - **max**: $s_{wt} = \max(r_{wt}, 1 - r_{wt})$;
     - **log**: $s_{wt} = \max\left(\ln \frac{r_{wt}}{1 - r_{wt}}, \ln \frac{1 - r_{wt}}{r_{wt}}\right)$;
     - **identity**: $s_{wt} = r_{wt}$.

- Estimation of the word distribution in topics ϕ_{wt} (**phi_init**):
  - **identity**: $\phi_{wt} \propto s_{wt}$;
  - **threshold**: $\phi_{wt} \propto [s_{wt} > \alpha]$.
- Estimation of the topic distribution in documents θ_{td} (**theta_init**):
  - **threshold**: by using the number of words with big importance estimate according to the formula (3);
  - **prob_log**: by using Bayesian probability estimate according to the formula (2).

In Bayesian methods, we considered the threshold α to be equal to $\{0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50, 100\}$. For each initialization option, a smoothing constant was chosen from the set $\{0.01, 0.1, 1, 3, 5, 10\}$. This constant was added to each element of the matrices Φ and Θ before the normalization of those matrices. The regularizing coefficient λ was chosen from the set $\{0, 10^{-6}, 10^{-4}, 10^{-2}, 10^{2}, 10^{4}, 10^{6}, 10^{8}\}$. As a result, we considered 2840 methods of the topic model adjustment per each data set.

We have chosen the coefficient of determination R^2 as a quality criterion:

$$R^2 = 1 - \frac{\sum_{d \in D}(a(d) - y_d)^2}{\sum_{d \in D}(y_d - \bar{t})^2},$$

where $a(d)$ is a linear model over the matrix Θ, adjusted according to the chosen method and $\bar{y}$ is the average value of the target value. This criterion was computed on test sets.

The results are presented in the table 1. Bayesian methods of initialization generally perform better, despite the K-Means method of the documents clustering gave the best quality for the Salary problem. The best method of initialization is probably the Bayesian one with $s_{wt} = r_{wt}$, $\phi_{wt} = [s_{wt} > \alpha]$ and computation of θ_{td} with the formula (2). Results of this method are among the four best ones for Movie Reviews, Salary and unregularized Yelp. With comparison to the random initialization, this methods improves quality by 130% and 32% for the problems MovieReviews and Salary respectively. MSE-regularisation (1) contributes essentially as well: thanks to it the quality rises up to 53% and 22% with comparison to the usual PLSA method of the same initialization.

4.2 Comparison with vector space models

Among relatively popular methods of solving regression and classification problems on texts there are various methods connected with the vector space models. The idea is to map each document d to a real-valued vector v_d of dimension K, where K is the vocabulary size. Elements of this vector may be computed in different ways — for instance, they may be equal to the numbers of occurrences of corresponding words to the document, or to the TF-IDFs of those words. These vectors can be then used as features for any machine learning method.

The comparison of the following methods of features generation was conducted:

1. **count**: a vector representation, elements of the vector are counters of the words occurrences to the document;

2. **tfidf**: a vector representation, elements of the vector are TF-IDFs of the words;

	MovieReviews	Salary	Yelp
slda	0.2280	0.1655	0.1783
theta_rand	0.1707	0.1986	0.1993
theta	0.4766	0.3663	0.4121
count	0.5537	0.2953	0.3525
tfidf	0.5599	0.4007	0.4627
count+theta_rand	0.5536	0.2958	0.3524
tfidf+theta_rand	0.5650	0.4018	0.4627
count+theta	0.5813	0.3438	0.4450
tfidf+theta	**0.5821**	**0.4187**	**0.4628**

Table 2: The coefficients of determination for the test sample for various options of features generation. The best result of each column is marked bold.

3. **slda**: aposteriori probabilities of the topics in documents from the sLDA [8] method are used as features;

4. **theta**: columns of the Θ matrix (topic distribution in documents) are used as features, the Bayesian initialization method and the MSE-regularization are used for adjusting the topic model;

5. **theta_rand**: similarly to the previous method, but with no initialization and regularization.

6. **count+theta**: words occurrences counters and the Θ matrix columns as well;

7. **tfidf+theta**: TF-IDFs of the words and the Θ matrix columns as well.

For construction of the Θ matrix for the Movie Review and Salary problems we used the initialization method `score_type = identity, phi_init = threshold, theta_init = prob_log` with thresholds 0.6 for Movie Review and 0.7 for Salary and initialization method `score_type = log, phi_init = threshold, theta_init = threshold` with the threshold 1 for Yelp.

We trained the linear regression with the L_2- or L_1- regularization (i.e. ridge regression and LASSO) on those features and chose the quality of the best method among those. The resuts are shown in the table 2.

TF-IDF guarantees very good quality for all of the problems, but we were able to improve this quality for Movie Review and Salary problems with a topic model. Note that a topic model produces 10 features only, but we can obtain decent quality with them; moreover, for the Salary and Yelp problems we were able to overcome the vector space model with word counters, despite it uses thousands of features. Thus, the topic model allows reduction of the feature space dimension by several orders of magnitude, simultaneously saving essential amount of information.

4.3 Multigrams approach

In this section we consider the influence of the multigram techniques on the topic model quality and compare them with the methods described above.

An n-gram is defined just as an n-tuple of consecutive words $(w_1, ..., w_n)$. It seems natural to try to use statistical information about n-grams in documents during the topic model adjustment. Indeed, in terms of the natural language, an occurrence of a word-combination `construction worker` means quite a lot for a job advertisement, while a simultaneous occurrence of its parts `construction` and `worker` is not

	MovieReviews		Salary		Yelp	
	no reg	mse reg	no reg	mse reg	no reg	mse reg
random	0.2065	0.2065	0.1641	0.2776	0.1432	0.1433
cluster	0.2139	0.4024	0.1971	**0.3722**	0.2354	0.3503
hash	0.1210	0.3542	0.2018	0.3186	0.1782	0.1792
random_topic	0.1889	0.2699	0.1938	0.3354	0.2231	**0.4093**
subsample-1	0.2005	0.3037	0.2056	0.3481	0.2297	0.3474
subsample-2	0.2239	0.3059	0.2282	**0.3700**	0.2485	0.2294
subsample-3	0.2283	0.3683	0.2311	0.3200	0.2365	0.3024
subsample-5	0.2167	0.3356	0.2077	0.3330	0.2210	0.3869
score_type=identity, phi_init=identity, theta_init=prob_log	0.2647	0.4018	0.2734	0.3265	**0.3211**	0.3359
score_type=identity, phi_init=threshold, theta_init=prob_log	**0.3109**	**0.4753**	**0.3002**	**0.3663**	**0.2984**	0.3118
score_type=identity, phi_init=threshold, theta_init=threshold	**0.2926**	0.4245	0.2655	0.3181	0.2608	0.3503
score_type=log, phi_init=identity, theta_init=prob_log	0.2671	0.4434	0.2826	0.3537	0.2221	0.2222
score_type=log, phi_init=threshold, theta_init=prob_log	**0.2982**	**0.4685**	**0.3159**	0.3633	0.2659	0.2660
score_type=log, phi_init=threshold, theta_init=threshold	0.2851	**0.4610**	**0.3099**	**0.3669**	0.2778	**0.4121**
score_type=max, phi_init=identity, theta_init=prob_log	0.2604	0.4111	**0.3075**	0.3331	**0.2790**	0.2790
score_type=max, phi_init=threshold, theta_init=prob_log	0.2742	**0.4706**	0.2906	0.3511	**0.3035**	**0.4002**
score_type=max, phi_init=threshold, theta_init=threshold	**0.2882**	0.4130	0.2815	0.3590	0.2703	**0.4008**

Table 1: Coefficients of determination on the test set for different initialization options. We consider the PLSA method with no regularization (no reg) and with MSE-regularization (mse reg) using the formula (1). The best four results in each column are marked bold.

as informative. This observation encourages us to try embedding those word-combinations in the same probabilistic context.

In the same time, the usage of all possible n-grams would lead us to a very complicated model inclined to overfitting. Hence, one needs to introduce a criterion for the multigrams selection. Among such criteria there are TF-IDF-based, CValue-based and frequency-based ones. For more details we refer to [19] and [20].

We have considered n-grams for $n = 2$ only. We have selected the optimal number of most frequent bigrams in corpus used as terms for each dataset. Moreover, we have collected the information about bigrams after the filtering, so we have, strictly speaking, dealt with *skip-grams*.

We have chosen the method `score_type = identity, phi_init = threshold, theta_init = prob_log` for the experiments with the bigrams, as the corresponding row in the table 1 contains many **bold** results.

It turned out that this approach had affected different datasets in different ways. We list the results (the predictive determination coefficient values) below.

Movie Review.

We did not manage to obtain any improvement on the MovieReviews data set. This fact can be explained by a small number of documents (4000 in the training set), so the extension of the features space leads to convergence or overfitting problems.

bigrams count	no reg	mse reg
0	0.3109	0.4753
15	0.3048	0.4133
50	0.2985	0.3930
100	0.2901	0.3888
2500	0.2658	0.1706

Salary.

We've found that it is possible to obtain decent results with multigrams techinques on this data set for non-regularized functional. Note that the quality gain is comparable to the one obtained by the regularization.

bigrams count	no reg	mse reg
0	0.3002	0.3663
100	0.3218	0.3210
125	0.3170	0.3553
150	0.3190	0.3577
1000	0.2821	0.3128

Yelp.

In the case of this data set, we were able to improve the quality of the non-regularized method; the results of regularized multigram method turned out to be inadequate.

bigrams count	no reg
0	0.2984
100	0.3001
150	0.3005
200	0.2977

We can conclude that adding multigrams to the topic models is a promising approach for quality improvements, but it definitely has some challenges and will be the subject of our further work.

5. CONCLUSION

A method for the topic models application to the regression problems was introduced. This method consists of two parts. At first, we initialize the topic model based on the naive Bayes classifier. The optimization method is able to find a decent local minimum thanks to this classifier. Then, we use the MSE-regularizer during the model adjustment. It requires that the target variable is expressed as a linear function of probabilities of topics in the document.

The real data experiments have shown that the initialization and the regularization both allow sufficient quality gain. Topic probabilities in the documents themselves are good features, they allow us to essentially reduce the feature space dimensionality with low quality loss. The usage of those probabilities in combination with TF-IDF features allows us to obtain the best quality among all of the methods used in the comparison.

Acknowledgements.

The work was supported by the Russian Foundation for Basic Research (14-07-31163).

6. REFERENCES

[1] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the ACL, 2005.

[2] D. Newman, C. Chemudugunta, P. Smyth, and M. Steyvers. Analyzing entities and topics in news articles using statistical topic models. In Intelligence and Security Informatics, Lecture Notes in Computer Science. 2006.

[3] T. Griffiths, M. Steyvers. Finding scientific topics. Proceedings of the National Academy of Sciences 101 (Suppl 1): 5228-35. 2004.

[4] M. Paul, R. Girju. Cross-Cultural Analysis of Blogs and Forums with Mixed-Collection Topic Models. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1408-1417, Singapore. 2009.

[5] D. Blei et. al. Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022. 2003.

[6] I. Porteous et. al. Fast Collapsed Gibbs Sampling For Latent Dirichlet Allocation. Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 569-577. 2008.

[7] D. Mimno, M. Hoffman, D. Blei. Sparse Stochastic Inference for Latent Dirichlet Allocation. In Proceedings of the International Conference on Machine Learning, Edinburgh, 2012.

[8] D. Blei, J. McAuliffe. Supervised topic models. Neural Information Processing Systems 21. 2007.

[9] Jun Zhu, Amr Ahmed, Eric P. Xing. MedLDA: Maximum Margin Supervised Topic Models for Regression and Classification. In ICML, Montreal, Canada. 2009.

[10] M. Rabinovich, D. Blei. The inverse regression topic model. International Conference on Machine Learning. 2014.

[11] D. Mimno, A. McCallum. Topic Models Conditioned on Arbitrary Features with Dirichlet-multinomial Regression. UAI, page 411-418. AUAI Press. 2008.

[12] M. Meila, D. Heckerman. An experimental comparison of several clustering and initialization methods. In Proc. of Uncertainty in Artificial Intelligence, UAI'98, p. 386–395. 1998.

[13] J. Blomer, K. Bujna. Simple Methods for Initializing the EM Algorithm for Gaussian Mixture Models. Computing Research Repository, http://arxiv.org/abs/1312.5946. 2013.

[14] K. Vorontsov, A. Potapenko. Additive Regularization of Topic Models. Machine Learning Journal, Special Issue "Data Analysis and Intelligent Optimization", Springer, 2014.

[15] T. Hofmann. Probabilistic Latent Semantic Analysis. In Proc. of Uncertainty in Artificial Intelligence, UAI'99, p. 289–296. 1999.

[16] Q. Shi et.al. Hash Kernels for Structured Data. Journal of Machine Learning Research, 10, 2615-2637. 2009.

[17] A. Agarwal, O. Chapelle, M. Dudik, J. Langford. A Reliable Effective Terascale Linear Learning System. Journal of Machine Learning Research, 15, 1111-1133. 2014.

[18] D. Arthur, S. Vassilvitskii. k-means++: The Advantages of Careful Seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, SODA'07, p. 1027–1035. 2007.

[19] R. Hussey, S. Williams, R. Mitchell. Automatic keyphrase extraction: a comparison of methods. In Proc. of the 4th International Conference on Information, Process, and Knowledge Management, p. 18-23. 2012.

[20] K. Frantzi, S. Ananiadou, H. Mima. Automatic recognition of multi-word terms: the C-value/NC-value method. Intl. J. of Digital Libraries Vol. 3 Issue 2, p. 117-132. 2000.